

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-272662  
(43)Date of publication of application : 18.10.1996

(51)Int Cl G06F 12/00

(21)Application number : 07-350677 (71)Applicant : MICROSOFT CORP  
(22)Date of filing : 13.12.1995 (72)Inventor : GUZAK CHRISTOPHER J  
NAKAJIMA SATOSHI

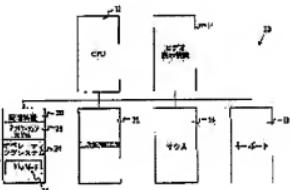
(30)Priority  
Priority number : 94 355397 Priority date : 13.12.1994 Priority country : US

#### (54) DATA TRANSFER WITH EXPANSION CLIPBOARD FORMAT

(57) Abstract:

**PROBLEM TO BE SOLVED:** To provide expansion clipboard formats which increase the number of formats usable for a clipboard in a computer system.

**SOLUTION:** A user in an application can expand a use of a clipboard and other data transfer mechanism by these expansion clipboard formats. The clipboard formats can be used by means of a conventional clipboard, a drag and drop mechanism and an OLE data transfer mechanism. Several expansion clipboards are adapted to be able to be used for data transfer of non file system objects.



(19) 日本国特許庁 (JP)

## (12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-272662

(43) 公開日 平成8年(1996)10月18日

(51) Int.Cl.<sup>6</sup>  
G 0 6 F 12/00識別記号  
5 1 5府内整理番号  
7623-5BF I  
G 0 6 F 12/00技術表示箇所  
5 1 5 M

審査請求 未請求 請求項の数33 O.L 外国語出願 (全 36 頁)

(21) 出願番号 特願平7-350677  
 (22) 出願日 平成7年(1995)12月13日  
 (31) 優先権主要番号 08/355397  
 (32) 优先日 1994年12月13日  
 (33) 优先権主要国 米国(US)

(71) 出願人 391055933  
 マイクロソフト コーポレイション  
 MICROSOFT CORPORATION  
 アメリカ合衆国 ワシントン州 98052-  
 6399 レッドモンド ワン マイクロソフ  
 ト ウエイ (番地なし)  
 (72) 発明者 クリストファー ジェイ グザーク  
 アメリカ合衆国 ワシントン州 98033  
 カークランド ノースイースト シックス  
 ティセカンド 10821  
 (74) 代理人 弁理士 中村 稔 (外7名)

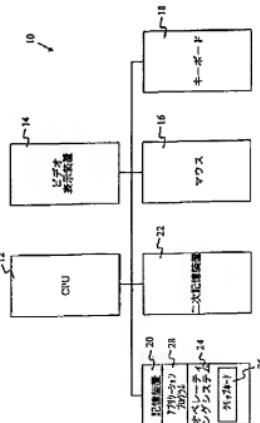
最終頁に続く

(54) 【発明の名稱】 拡張クリップボードフォーマット付データトランസ്ഫ

## (57) 【要約】

【課題】 コンピュータシステムは、クリップボードで使用できるフォーマット数を増大する拡張クリップボードフォーマットを提供する。

【解決手段】 これらの拡張クリップボードフォーマットにより、アプリケーションにおけるユーザは、クリップボードおよび他のデータ転送機構の使用を広げることができる。クリップボードフォーマットは、従来のクリップボードにより、ドラッグ・アンド・ドロップ機構により、およびOLE データ転送機構により利用できる。幾つかの拡張クリップボードフォーマットは、非ファイルシステム対象物のデータ転送に使用できるように適応される。



## 【特許請求の範囲】

【請求項1】 クリップボードフォーマットにおけるデータのデータ転送を実施するクリップボードを有するコンピュータシステムにおいて、ファイルの内容を保持するファイル内容クリップボードフォーマットを形成するステップと、データ転送用のファイルでないデータであって、ファイル内容クリップボードフォーマット内にあるデータを提供するステップと、およびそのデータについてデータ転送を実施して、データを保持するファイルを生成するステップとから構成される方法。

【請求項2】 データ転送を実施するステップが、クリップボードを使用し、データ転送を実施して、データを保持するファイルを生成するステップから成る請求項1の方法。

【請求項3】 データ転送を実施するステップが、ドラグ・アンド・ドロップ操作を実施し、データ転送を実施して、データを保持するファイルを生成するステップから成る請求項1の方法。

【請求項4】 データは複合文書内の埋め込み部分である請求項1の方法。

【請求項5】 (a) ファイルの内容を保持するファイル内容クリップボードフォーマットを形成する手段と、(b) ファイルでないデータを提供するデータ源と、(c) 転送されたデータを受け入れるデータシンクと、(d) ファイル内容クリップボードフォーマット内のデータ源からデータシンクへデータを転送して、データを保持するファイルを生成するデータ転送機構とから構成されるコンピュータシステム。

【請求項6】 データ転送を実施するクリップボードを有するコンピュータシステムにおいて、

ファイルグループ記述子を保持するファイルグループ記述子クリップボードフォーマットを形成するステップと、データ転送のためのデータのグループ用のファイルグループ記述子であって、データのグループについての記述情報を含むファイル記述子を保持する前記ファイルグループ記述子はファイルではなく、また前記ファイルグループ記述子はファイルグループ記述子クリップボードフォーマット内にあるようにしたファイルグループ記述子を提供するステップと、

グループデータのデータ転送を実施するステップとから構成される方法。

【請求項7】 データ転送を実施するステップが、データ転送を実施してファイルのグループを生成するステップから成り、また各ファイルがグループでのデータと関連している請求項6の方法。

【請求項8】 ファイルの内容を保持するファイル内容クリップボードフォーマットを形成するステップ、およびファイル内容クリップボードフォーマットにおけるグ

ループでのデータを提供するステップからさらに構成される請求項7の方法。

【請求項9】 データ転送を実施するステップが、クリップボードを使用してデータ転送を実施するステップから成る請求項6の方法。

【請求項10】 データ転送を実施するステップが、ドラグ・アンド・ドロップ操作を実施してデータ転送を実施するステップから成る請求項6の方法。

【請求項11】 (a) ファイルグループ記述子を保持するファイルグループ記述子クリップボードフォーマットを形成する手段と、

(b) ファイルグループ記述子クリップボードフォーマットにおいてファイルグループ記述子を提供するデータ源において、前記ファイルグループ記述子が、ファイルではないデータのグループについての記述情報を有するファイル記述子を保持するようにしたデータ源と、

(c) ファイルグループ記述子クリップボードフォーマットにおいて転送されたデータを受け入れるデータシンクと、

(d) ファイルグループ記述子をデータシンクへ転送するデータ転送機構とから構成されるコンピュータシステム。

【請求項12】 (e) ファイルの内容を保持するファイル内容クリップボードフォーマットを形成する手段からさらに構成される請求項11のコンピュータシステム。

【請求項13】 データ源が、保持するファイル内容クリップボードフォーマットにおいてデータのグループを提供する手段からさらに構成され、またデータ転送機構が、データ源からデータシンクへファイル内容クリップボードフォーマットにおけるファイルのグループを転送してファイルのグループを生成する手段からさらに構成される請求項12のコンピュータシステム。

【請求項14】 データ転送機構がクリップボードから成る請求項11のコンピュータシステム。

【請求項15】 クリップボードフォーマットにおけるデータのデータ転送を実施するクリップボードを有するコンピュータシステムにおいて、データ転送操作に関与するファイルのリストを記述するファイルリスト構成へアクセスする値を格納するファイルリスト・クリップボードフォーマットを形成するステップと、

ファイルリスト・クリップボードフォーマットにおいてデータを提供するステップと、ファイルリスト・クリップボードフォーマットにおけるデータについてデータ転送を実施するステップとから構成される方法。

【請求項16】 ファイルリスト・クリップボードフォーマットにおいてデータを提供するステップが、ファイルリスト・クリップボードフォーマットにおけるファイルにアクセスする値を提供するステップから成る請求項

15の方法。

【請求項1 7】 ファイルリスト・クリップボードフォーマットにおいてデータを提供するステップが、ファイルのリストの識別子を提供するステップから成り、前記識別子がファイルリスト・クリップボードフォーマット内にある請求項15の方法。

【請求項1 8】 データ転送を実施するステップが、クリップボードを使用して、ファイルリスト・クリップボードフォーマットにおけるデータについてデータ転送を実施するステップから成る請求項15の方法。

【請求項1 9】 データ転送を実施するステップが、ドラグ・アンド・ドロップ操作を実施して、ファイルリスト・クリップボードフォーマットにおけるデータについてデータ転送を実施するステップから成る請求項15の方法。

【請求項2 0】 (a) データ転送操作に固有するファイルのリストを記述するファイルリスト構成へアクセスする値を格納するファイルリスト・クリップボードフォーマットを形成する手段と、

(b) ファイルリスト・クリップボードフォーマットにおいてデータを提供するデータ源と、

(c) ファイルリスト・クリップボードフォーマットにおいてデータを受け入れるデータシンクと、

(d) データ源からデータシンクへデータを転送するデータ転送機構とから構成されるコンピュータシステム。

【請求項2 1】 クリップボードフォーマットにおいてデータのデータ転送を実施するクリップボードと出力装置を有するコンピュータシステムにおいて、图形対象物を出力装置上に表示するときに、图形対象物の互いの相対位置を格納する対象物位置クリップボードフォーマットを形成するステップと、  
間連する图形対象物を有する対象物を提供するステップと、

対象物位置クリップボードフォーマットにおいて出力装置上に表示される图形対象物の現在位置についての相対対象物位置を提供するステップと、

対象物および相対対象物位置のデータ転送を実施して、同一の相対対象物位置で、新しい位置における图形対象物を出力装置上に表示するステップとから構成される方法。

【請求項2 2】 データ転送がクリップボードにより実施される請求項2 2 [sic] の方法。

【請求項2 3】 (a) 幾つかの対象物を提供するデータ源と、

(b) データ源により提供される対象物を表す图形対象物を表示する出力装置と、

(c) 出力装置上の图形対象物のグループの相対位置を格納する対象物位置クリップボードフォーマットを形成する手段と、

(d) データ源からデータシンクへデータを転送し、かつ

対象物位置クリップボードフォーマットにおける图形対象物の相対位置をデータ対象物から受信して、転送が完了すると、图形対象物の相対位置が保全されるようにしたデータ転送機構とから構成されるコンピュータシステム。

【請求項2 4】 クリップボードフォーマットにおいてデータのデータ転送を実施するクリップボードと出力装置を有するコンピュータシステムにおいて、データ転送を通して転送される項目用の少なくとも1つの代替名のリストを保持するファイル名マップ・クリップボードフォーマットを形成するステップと、

ファイル名マップ・クリップボードフォーマットにおける項目のリストの各項目について代替名のリストを提供するステップにおいて、項目のリストが、データのリストについてデータ転送を実施する少なくとも1つの項目を含み、その項目には、データ転送が完了すると、その代替名が割り当てられるようにしたステップとから構成される方法。

【請求項2 5】 データ転送がクリップボードにより実施される請求項24の方法。

【請求項2 6】 データ転送がドラグ・アンド・ドロップ操作により実施される請求項24の方法。

【請求項2 7】 (a) 項目のリストにおける各項目について代替名を保持するファイル名マップ・クリップボードフォーマットを形成する手段と、

(b) 転送される少なくとも1つの項目のリストを提供するデータ源と、

(c) 代替名が各項目毎に提供されるように、データ源により提供される項目のリストについて代替名のリストを提供する手段において、代替名の前記リストが、ファイル名マップ・クリップボードフォーマットにあるようにした手段と、

(d) 転送される項目のリストを受信するデータシンクと、

(e) リストの各項目に、ファイル名マップ・クリップボードフォーマットにある代替名のリストにおいて提供される代替名が割り当てられるように、データ源からデータシンクへ項目のリストを転送するデータ転送機構とから構成されるコンピュータシステム。

【請求項2 8】 ファイルシステム対象物および非ファイルシステム対象物を含む対象物のネーム空間、およびクリップボードフォーマットにおいてデータを転送するクリップボードとを有するコンピュータシステムにおいて、

非ファイルシステム対象物へアクセスできるデータを保持する選択されたクリップボードフォーマットを形成するステップと、

選択されたクリップボードフォーマットにおける非ファイルシステム対象物の一定の1つにアクセスできるデータを提供するステップと、

5

選択されたクリップボードフォーマットにおけるデータを使用して、一定の非ファイルシステム対象物を転送するためにデータ転送を実施するステップとから構成される方法。

【請求項29】 一定の非ファイルシステム対象物が、ネットワーク資源である請求項25の方法。

【請求項30】 一定の非ファイルシステム対象物がプリンタである請求項28の方法。

【請求項31】 (a) ファイルシステム対象物および非  
ファイルシステム対象物と、

(b) ファイルシステム対象物および非ファイルシステム対象物用のネーム空間を提供するネーム空間機構と、  
(c) 非ファイルシステム対象物へアクセスできるデータを保持する選択されたクリップボードフォーマットを形成する手段と、

(d) 一定の非ファイルシステム対象物へアクセスできるデータを使用して、非ファイルシステム対象物の一定の1つを転送するデータ転送機構において、そのデータが選択されたクリップボードフォーマットにあるようにした機構とから構成されるコンピュータシステム。

【請求項32】 クリップボードフォーマットにおけるデータを転送するクリップボード、および複数の対象物を有するコンピュータシステムであって、各対象物が、それを識別する付属のIDを有し、またIDはIDリストに集約できるコンピュータシステムにおいて、

対象物のグループのIDリストを提供するステップにおいて、  
IDリストを保持するIDリスト・クリップボードフォーマットを形成するステップと、

て、前記IDリストが、対象物のそれぞれのIDを保持し、かつIDリスト・クリップボードフォーマットにあるようにしたステップと、

データ転送機構とIDリスト・クリップボードフォーマットにおけるIDリストを使用して対象物のグループを転送するステップとから構成される方法。

【請求項33】 (a) IDを有する対象物と、  
(b) 対象物の幾つかに対するIDのリストを有するIDリスト

トを保持するIDリスト・クリップボードフォーマットを形成する手段と、

(c) 対象物のグループ用のリスト・クリップボードオーマットにおけるIDリストを提供するデータ源と、  
(d) IDリスト・クリップボードオーマットにおけるデータ源により提供されるIDリストについてデータ転送を実施するデータ転送機構とから構成されるコンピュータ

ラスラム。

光明の世界

【産業上の利用分野】本発明は、一般にデータ処理システムに関し、特にデータ処理システムにおけるデータ転送に関する。

[0002]

【従来の技術】クリップボードは、マイクロソフト社(ワシントン州、レッドモンド)が販売している「マイクロソフト」ウインドウズ、バージョン3.1のオペレーティングシステムのデータ転送機能である。クリップボードは、アプリケーション間またはアプリケーション内でデータを転送するのに使用される。クリップボードは、機能とメッセージの集合を備えるので、アプリケーションはデータをクリップボードを通して転送できる。

10 物の独自の識別子)を格納する共通領域とみなすことができ、そのデータ名に基づいてアプリケーションは書式化されたデータを交換できる。「マイクロソフト」ウィンドウズ、バージョン3.1のオペレーティングシステムでは、一定数のクリップボードフォーマットを列挙する。例えば`CF_BITMAP`クリップボードはビットマップの転送に使用され、また`CF_TEXT`クリップボードはテキスト文字の配列の転送に使用される。クリップボードは、種々のクリップボードフォーマットで同一データを同時に保

20 り)、「複写」または「貼り付け」などのクリップボードコマンドを呼び出でてクリップボードを使用する。複写クリップボードコマンドは、クリップボードフォーマット内のデータの選択された部分を保持するデータ対象物の名前をクリップボードへ複写する点により、クリップボードフォーマット内のデータの選択された部分をクリップボードへ複写する。データの選択された部分は、そのデータの発信元から削除されない。切り取りクリップボードコマンドは、それがデータの選択された部分をクリップボードへ複写する点で複写コマンドと異なる。貼り付けクリップボードコマンドは、それがデータの選択された部分をデータの発信元から削除する点で複写クリップボードコマンドと異なる。貼り付けクリップボードコマンドは、クリップボードからデータを送信元へ複写する。

[0003]

【秦明が解決しようとする課題】マイクロソフト社が作成したマイクロソフトOLE 2.0!プロトコルは、ドラッグアンド・ドロップ操作を容易にする機能を提供する。この機能はクリップボードフォーマットを使用する。しかしながらデータは、切り取りや複数の操作に使用されるシステム記憶装置を通過しないで、代わりにデータは直接、元から送信先へ転送される。クリップボードは有用であるが、書式化の制約により、多くのアプリケーションが過大に限定されている。特に、そのシステムにより提供されるクリップボードフォーマットは少な過ぎ、さらに利用できるフォーマットは過去に限られている。

[0 0 0 4]

【課題を解決するための手段】本発明は、拡張クリップボードフォーマットを提供することにより、従来のシステムの制約を克服する。これらの拡張クリップボードフ

オーマットは、ファイルの内容を保持するクリップボードフォーマットを備えるので、ファイルでないデータを転送でき、また転送の一部としてデータはファイル中に転送できる。また拡張クリップボードフォーマットは、ファイルグループ記述子を保持するクリップボードフォーマットを備えることができる。このファイルグループ記述子は幾つかのファイル記述子を保持し、また各ファイル記述子は、ファイルについて、またはデータ転送操作中にファイルに組み込まれることになるデータについての記述情報を保持する。本発明の実施例により提供される拡張クリップボードフォーマットは、ファイルリスト構造へアクセスする値を格納するファイルリスト・クリップボードフォーマットを備えることができる。ファイルリスト構造は、ファイルのリストを記述する。このファイルリスト・クリップボードフォーマットは、他のクリップボードフォーマットのように、データ転送操作中に使用できる。

【0005】また拡張クリップボードフォーマットは、图形対象物が出力装置上に表示されるときに图形対象物の相対位置を格納する対象物位置クリップボードフォーマットを備えることができる。このクリップボードフォーマットは、データ転送操作が完了した後に対象物に付随する图形対象物の相対的な対象物位置を保存するため、対象物のデータ転送に使用できる。また拡張クリップボードフォーマットは、データ転送操作中に転送されている項目用の代替名のリストを保持するファイル名マップ・クリップボードフォーマットを備えることができる。ファイル名マップ・クリップボードフォーマットに保持される代替名のリストは、データ転送が一旦完了すると、ファイルを改名するのに使用できる。本発明の実施例により提供される拡張クリップボードフォーマットは、非ファイルシステム対象物へアクセスできるデータを保持する、選択されたクリップボードフォーマットを備えることができる。一定のコンピュータシステムは、共通ネーム空間内にファイルシステム対象物と非ファイルシステム対象物とを備えることができる。この拡張クリップボードフォーマットは、選択されたクリップボードフォーマットの非ファイルシステム対象物のデータ転送を容易にする。非ファイルシステム対象物は、ネットワーク資源、プリンタまたは他の形式の対象物を含むことができる。

【0006】本発明の実施例により提供される拡張クリップボードフォーマットは、対象物のIDリストを保持するIDリストクリップボードフォーマットを備えることができる。このクリップボードフォーマットは、IDリストに集約される関連するIDを保持する対象物を有するコンピュータシステムにおいて有用である。IDリスト・クリップボードフォーマットは、対象物がデータ転送機構を使用して転送される場合に有用である。本発明の実施例は、添付図面を参照して詳細に説明される。

## 【0007】

【実施例】本発明の好ましい実施例は、拡張された数のクリップボードフォーマットを提供する。これらの拡張クリップボードフォーマットは、従来のシステムにより得られるものより改良された機能性を提供するクリップボードフォーマットを備える。図1は、本発明の好ましい実施例を実施するのに適したコンピュータシステム10のブロック図である。コンピュータシステム10は、ビデオ表示装置14、マウス16およびマウスボード18のような幾つかの周辺装置-マウスする中央処理装置(CPU)12を備える。CPU 12は、一次記憶装置20および二次記憶装置22へもアクセスする。一次記憶装置20は、クリップボード26を備えるオペレーティングシステム24を保持する。クリップボード26は、データ転送を実現するクリップボード能力を提供する機能およびメッセージ、と本発明の好ましい実施例を実施するクリップボードフォーマットとを含む。また一次記憶装置20は、少なくとも1つの中アリケーションプログラム28用のコードを保持する。技術に有能な者にとり、図1のコンピュータシステム10がもっぱら説明上のものであることは明白である。

本発明は、多重プロセッサを採用するもの、および図1に示す装置とは異なる周辺装置を採用するものを含む、他のコンピュータシステム構成においても実施できる。

【0008】以下の説明は、本発明の好ましい実施例により提供される拡張クリップボードフォーマットおよびそれらの使用を特に列挙したものである。これらの拡張クリップボードフォーマットは、「マイクロソフト」ウインドウズ、バージョン3.1のオペレーティングシステムにより提供されるものに成って代わるのではなく、それを補足することができる。図2は、好ましい実施例の拡張クリップボードフォーマットを利用するために実施されるステップを図示するフローテーブルである。先ず、アリケーションプログラム28は、データの選択された部分を1つ以上の拡張クリップボードフォーマットに変換しなければならない(ステップ30)。このアリケーションプログラムは、拡張クリップボードフォーマットに変換されるデータを識別する機能を提供する。変換されたデータは、データ対象物に格納される(ステップ32)。データ対象物は、大域記憶装置、ストリームまたは記憶機械のような種々の形式の記憶媒体(例えば「マイクロソフト」OLE 2.0)に存在できる。拡張クリップボードフォーマットの幾つかに對するデータ対象物は、以下の詳細に説明するように、考えられる記憶媒体の部分集合だけに存在できる。データ構造を封入するデータ対象物の名前は、クリップボードに渡される(ステップ34)。ついでデータは、拡張クリップボードフォーマット内のクリップボード上にあるとみなされ、ユーザまたはアリケーションプログラムが必要とするときに送信先へ貼り付けられる(ステップ36)。

【0009】本発明の好ましい実施例により提供される

拡張クリップボードフォーマットの幾つかに對するデータ対象物は、以下の詳細に説明するように、考えられる記憶媒体の部分集合だけに存在できる。データ構造を封入するデータ対象物の名前は、クリップボードに渡される(ステップ34)。ついでデータは、拡張クリップボードフォーマット内のクリップボード上にあるとみなされ、ユーザまたはアリケーションプログラムが必要とするときに送信先へ貼り付けられる(ステップ36)。

【0010】本発明の好ましい実施例により提供される

拡張クリップボードフォーマットの1つは、**CF\_HDROP** クリップボードフォーマットである。このクリップボードフォーマットの役割の説明には、「マイクロソフト」ウインドウズ、バージョン3.1 のオペレーティングシステムにおいて、ドラグ・アンド・ドロップ操作がどのようにして実施されるかを見直すのが役立つ。この従来のオペレーティングシステムにおいて、ドラグ・アンド・ドロップ操作は、マウスを使用するユーザに応答して実施される。マウスボタンを離してドロップを実施すると、マウスカーソルが現在指示しているウインドウは、**WM\_DROPFILES** ウィンドウメッセージを受信する。このウインドウメッセージには単一のパラメータである**hDrop** \* 10 \*p が含まれる。hDrop パラメータは、ドロップされたファイルまたは複数のファイルを記述するデータ構造の名前である。この名前は、hDrop により識別されるデータ構造から情報を検索する定義済の**DragQueryPoint()**、**DragQueryFile()** および**DragFinish()**機能への呼び出しにおいてパラメータとして使用される。**CF\_HDROP** クリップボードフォーマットは、**DROPPFILES** 構造の名前を保持するのに使用される。**CF\_HDROP** クリップボードフォーマットを使用すると、データ対象物(すなわち**DROPPFILE** 10)が大域記憶装置に存在することが必要となる。**DROPPFILE** 構造は下記のフォーマットを有する。

```
typedef struct _DROPPFILES {
    WORD pFiles;           // offset of file list
    POINT pt;              // drop point (client coords)
    WORD fNC;              // is it on non client area
    // and pt is in screen coords
    BOOL fWide;             // Wide character switch
} DROPPFILES, *LPDROPPFILES;
```

**fWide** フィールドは、**pFiles**により参照される文字列が ANSIではなくUNICODEで表されることを示している。**DragQueryPoint()**機能、**DragQueryFile()**機能および**DragFinish()**機能の全ては、**DROPPFILES**構造へ作用して、ドロップされたファイルについての情報を検索できる。**CF\_HDROP** クリップボードフォーマットを使用することにより、上述の機能は、クリップボード操作およびOLE データ転送操作のために働くことができる。従来のシステムにおける、これらの機能はドラグ・アンド・ドロップ状態においてだけ働く。

【0011】本明発の好ましい実施例の拡張クリップボードフォーマットは、**CF\_FILECONTENTS**クリップボードフォーマットおよび**CF\_FILEGROUPDESCRIPTR**クリップボードフォーマットも備える。**CF\_FILECONTENTS**クリップボードフォーマットは、ファイル中に封入されるデータを保持するのに使用される。例えばユーザが、私信メッセージまたは他の複合文書から埋め込み部分をドラッグし、ついでそれをファイルを生成する位置にドロップしたいとする。そのような場合、図3のフローチャートに示されるステップが実施される。先ず、転送されるデータが選択される(ステップ37)。そのデータは、**CF\_FILECONTENTS**クリップボードフォーマットに変換されて、格納される(ステップ38)。ついでデータ転送操作が完了し(ステップ39)、ドラグ・アンド・ドロップの目標対象物が、**CF\_FILECONTENTS**クリップボードフォーマットに格納されるデータにアクセスする。ついで**CF\_FILECONTENTS**クリップボードフォーマット内のデータは、ファイル中に封入される(ステップ40)。この例はドラグ・アンド・ドロップであったが、**CF\_FILECONTENTS**は、他の形式のデータ転送操作に使用できることが明らかである。クリップボードフォーマットがマイクロソフト OLE

20 E 2.01プロトコルに従うデータ転送操作において機能する役割を明らかにするには、OLE 2.01の幾つかの基本的な概念を見直すことが役立つ。OLE 2.01における「対象物」は、データを保持するデータ構造を含む論理構造であり、またデータ構造において保持されるデータについて働く機能を含むことができる。

【0012】OLE 2.01にとり基本的なものである他の概念は、「インタフェース」という概念である。インターフェースは、論理的に関連する機能のネーム集合である。インターフェースは、機能の集合に対して(パラメータのようない)指標を列挙する。インターフェースは、機能を実施するコードを提供するのではなく、むしろ、機能を実施するコードが対象物により提供される。インターフェースの機能を実施するコードを提供する対象物は、インターフェースを「サポート」すると言われる。インターフェースをサポートする対象物により提供されるコードは、インターフェースに設けられる指標に合致する必要がある。

C++ プログラム言語においてインターフェースは、仮想機能の集合を構成する。マイクロソフトOLE 2.01プロトコルは、転送データへの対象物にとり有用である幾つかの機能をグループ分けする**IDataObject**インターフェースを定義する。**IDataObject**インターフェースをサポートする対象物は、データ対象物として知られている。OLE 2.01を使用してデータ転送を実施するシステムにおいて、データが原始対象物から目標対象物へドラグされるドラグ・アンド・ドロップ操作中に、1994年2月22日に出願され、本出願と共通の譲受人に譲渡された「均等データ転送」という名称の同時係属出願第08/198, 853号に明示される均等データ転送機構が利用される。データ源におけるデータ対象物と目標対象物との間の接続は、データ対象物により提供される**IDataObject**インターフェースの事

例に対するポインタを送信先対象物へ渡すことにより、形成される。ついで送信先対象物は、`IDataObject` インターフェースの`EnumFormatEtc()`を実施して、送信先対象物上にドロップされた対象物におけるデータ用の利用できるフォーマットを列举する。これらのフォーマットはクリップボードフォーマットである。かくして、ドラグ・アンド・ドロップ操作中に、データはクリップボードフォーマットで転送される。

【0013】`CF_FILEGROUPDESCRIPTOR` クリップボードフォーマットは、ファイルグループ記述子構造を保持するのに使用される。ファイルグループ記述子構造は、1つ以上のファイル記述子を保持する。各ファイル記述子は、`CF_FILECONTENTS`クリップボードフォーマットで保持されるファイル中に封入されるデータについての情報を保持する構造体の配列である。各ファイル記述子は下記のフォーマットを有する。

```
typedef struct _FILEDESCRIPTOR { // fd
    DWORD dwFlags;
    CLSID cSID;
    SIZEL sizeL;
    POINTL pointL;

    DWORD dwFileAttributes;
    FILETIME ftCreationTime;
    FILETIME ftLastAccessTime;
    FILETIME ftLastWriteTime;
    DWORD nFileSizeHigh;
    DWORD nFileSizeLow;
    CHAR cFileName[MAX_PATH];
} FILEDESCRIPTOR, *LPFILEDESCRIPTOR;
```

`dwFlags` フィールドは、どのフィールドが正当なデータを保持するかを示すフラグであり、`cSID` フィールドは、`CF_FILECONTENTS`データを封入するデータ対象物のクラスIDを保持し、`sizeL` フィールドは、`CF_FILECONTENTS`データのサイズを規定する値を保持し、また`pointL` フィールドは、`CF_FILECONTENTS`データを保持する対象物へのポインタを保持する。`dwFileAttributes`フィールドは、データの属性のダブルワードを保持する。`ftCreationTime`フィールドは生成時間を持し、`ftLastAccessTime`フィールドは最後のアクセス時間を持し、また`ftLastWriteTime`フィールドは最後の書き込み時間を保持する。`nFileSizeHigh` フィールドは、対象物の長さをバイト単位で記述する64ビットの上位32ビットを保持し、また`nFileSizeLow` フィールドは、この値の下位32ビットを保持する。`cFileName` フィールドは、ファイル名を規定する文字を保持する。

【0014】`CF_FILEGROUPDESCRIPTOR` クリップボードフォーマットは、1つのパッチでの多重ファイルの転送を容易にする。転送されるパッチ用の、パッチにおけるファイルグループ記述子は、`CF_FILEGROUPDESCRIPTOR`

クリップボードフォーマットに格納される。ファイルグループ記述子は、大域記憶装置内にあり、他の原始媒體内にあってはならない。このファイルグループ記述子により、プログラムは、多重ファイル中へ転送されるデータをウォータースルーでき、また必要に応じて所要のファイルを生成できる。図4は、`CF_FILEGROUPDESCRIPTOR` クリップボードフォーマットと連係する`CF_FILECONTENTS`クリップボードフォーマットを使用して、データのグループを転送してファイルのグループを生成するために実施されるステップを図示するフローチャートである。

ファイル中に封入されるデータは先ず、変換されて、`CF_FILECONTENTS`クリップボードフォーマットに格納されなければならない（ステップ41）。ついで、データのグループ用のファイルグループ記述子は、`CF_FILEGROUPDESCRIPTOR` クリップボードフォーマットに格納される（ステップ42）。データは転送されて、ファイルのグループ中に封入される（ステップ43）。

【0015】`CF_FILECONTENTS`クリップボードフォーマットで転送されるデータを保持するデータ対象物は、大域記憶装置内に存在する対象物でもよいし、またはストリームや記憶装置のようなOLE 記憶媒体を通して提供してもよい。`CF_FILENAMEMAP` クリップボードフォーマットは、本発明の好ましい実施例により提供される拡張クリップボードフォーマットの他の1つのものである。このクリップボードフォーマットは、発信先名へのマッピングのための記憶装置を提供するのに使用される。例えばシステムが、ファイルがウエストバスケット・ファシリティに置かれるとき、ファイル名を変更したいとする。そのような場合、発信先名へのマッピングは`CF_FILENAMEMAP` クリップボードフォーマットに格納される。

このクリップボードフォーマットは、`CF_HDROP`クリップボードフォーマットの機能とみなすことができる。図5は、このクリップボードフォーマットを利用してファイル名を変更するため実施されるステップを図示するフローチャートである。先ず送信先名マッピングは、`CF_FILENAMEMAP` クリップボードフォーマットに入れられる（ステップ44）。ついで、データ転送は、クリップボードコマンド、ドラグ・アンド・ドロップまたは他のOLE データ転送機構を通して開始される（ステップ46）。

40 データ転送操作の完了の一環として、`CF_FILENAMEMAP` クリップボードフォーマットに保持されるデータ内に規定された新しいファイル名は、転送されているファイルに割り当たられる（ステップ48）。

【0016】拡張クリップボードフォーマットは、`CF_OBJECTPOSITIONS` クリップボードフォーマットも備える。このクリップボードフォーマットは、クリップボードにおける他の項目の位置に対する座標の配列を保持するのに使用される。座標の第1の集合は、対象物のグループの画面位置を識別し、また残りの座標は、画面単位での各項目の相対オフセットを規定する。このクリッ

13

ブードフォーマットは、関連するアイコンを有する対象物のグループを転送するのに特に有用である。このクリップボードフォーマットにより、データ変換が新しい送信先へ実施された後でも、項目のグループの相対位置の保全を同一のままにできる。図6は、**CF\_OBJECTPOSITIONS**クリップボードフォーマットを利用するために実施されるステップを図示するフローチャートである。まずアイコンは、**CF\_OBJECTPOSITIONS**クリップボードフォーマットに従って符号化される(ステップ50)。ついでデータ転送は、クリップボード、ドラッグ・アンド・ドロップまたは他のOLEデータ転送機構を通して開始される(ステップ52)。データ転送操作の完了の一部として、クリップボードフォーマットに保持される座標は、送信先においてアイコンを該当する相対位置にするのに使用される(ステップ54)。

【0017】拡張クリップボードフォーマットは、プリンタに好都合な名称のリストの名前を保持する**CF\_PRINTERFRIENDLYNAME**クリップボードフォーマットをさらに備える。ついでプリンタに好都合な名称は、対応するプリンタについて保持されるデータ構造へアクセスするのに使用できる。**CF\_NETRESOURCE**クリップボードフォーマットは、両振であるが、ネットワークサーバのようネットワーク資源のリストの名前を保持する。ネットワーク資源のリストは、資源用のデータ構造へアクセスするのに使用できる。このクリップボードフォーマットは、ネーム空間がファイルシステム構造ばかりでなく、他の形式の対象物を含むことができる状態の場合に特に使用される。**CF\_IDLIST**クリップボードフォーマットは、IDリストの名前を保持する。IDリストは、ネーム空間内のものを独自に識別する識別子のリストである。ID

リストは、ファイルまたはディレクトリ用のパスネームにはば類似するとみなすことができる。このクリップボードフォーマットは、**CF\_HDROP**クリップボードフォーマットとみなすことができる。本発明を、その好ましい実施例を参照して説明してきたが、技術に有能な者は、付記される特許請求の範囲により明示される本発明の範囲から逸脱することなく、詳細な方式での種々の変更ができるることは、技術に当業者にとり明白である。

## 【図面の簡単な説明】

【図1】本発明の好ましい実施例を実施するのに適切なコンピュータシステム上のブロック図である。

【図2】本発明の好ましい実施例の拡張クリップボードフォーマットを活用するように実施されるステップを図示するフローチャートである。

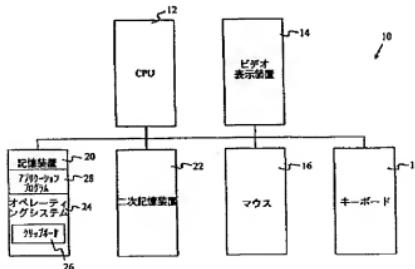
【図3】本発明の好ましい実施例に従うデータ転送操作において**CF\_FILECONTENTS**クリップボードフォーマットを使用するため実施されるステップを図示するフローチャートである。

【図4】本発明の好ましい実施例に従う**CF\_FILECONTENTS**および**CF\_FILEGROUPDESCRIPTR**クリップボードフォーマットの組合せ使用を図示するフローチャートである。

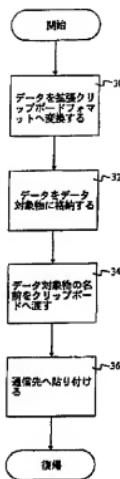
【図5】本発明の好ましい実施例に従う**CF\_FILENAMEMAP**クリップボードフォーマットの代表的な使用例において実施されるステップを図示するフローチャートである。

【図6】本発明の好ましい実施例に従う**CF\_OBJECTPOSITIONS**クリップボードフォーマットの代表的な使用例において実施されるステップを図示するフローチャートである。

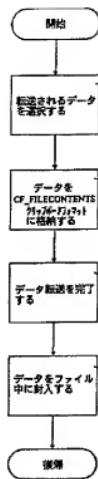
【図1】



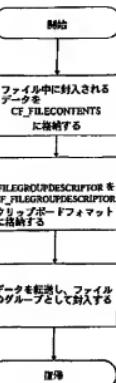
【図2】



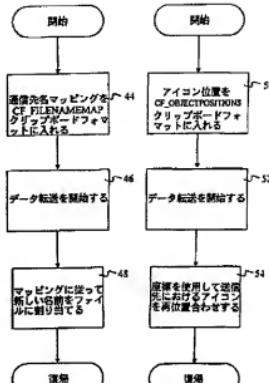
【図3】



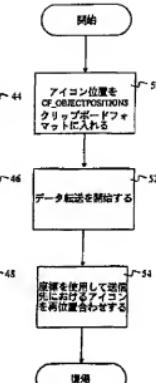
【図4】



【図5】



【図6】



フロントページの続き

(72)発明者 サトン ナカジマ

アメリカ合衆国 ワシントン州 98008

ベルヴィイー ワンハンドレッドアンドシ

ックスティエイス アベニュー ノースイ

ースト 3023

【外国語明細書】

## 1. Title of invention

DATA TRANSFER WITH EXPANDED  
CLIPBOARD FORMATS

## 2. Claims

1. In a computer system having a clipboard for performing data transfers of data in clipboard formats, a method comprising the steps of:

establishing a file contents clipboard format for holding contents of a file;  
providing data that is not a file for data transfer, said data being in the file contents clipboard format; and  
performing a data transfer on the data to create a file that holds the data.

2. The method of claim 1 wherein the step of performing the data transfer comprises the step of using the clipboard to perform the data transfer to create a file that holds the data.

3. The method of claim 1 wherein the step of performing the data transfer comprises the step of performing a drag-and-drop operation to perform the data transfer to create a file that holds the data.

4. The method of claim 1 wherein the data is an embedding in a compound document.

5. A computer system comprising:

(a) means for establishing a file contents clipboard format for holding contents of a file;  
(b) a data source for providing data that is not a file;  
(c) a data sink for accepting transferred data; and  
(d) a data transfer mechanism for transferring the data from the data source in the file contents clipboard format to the data sink to create a file holding the data.

6. In a computer system having a clipboard for performing data transfers, a method comprising the steps of:

establishing a file group descriptor clipboard format for holding a file group descriptor;  
providing a file group descriptor for a group of data for data transfer, said file group descriptor holding file descriptors that contain descriptive information about the group of data that are not files and said file group descriptor being in the file group descriptor clipboard format; and  
performing a data transfer of the group data.

7. The method of claim 6 wherein the step of performing the data transfer comprises the step of performing the data transfer to create a group of files, each file being associated with data in the group.

8. The method of claim 7, further comprising the steps of:

establishing a file contents clipboard format for holding contents of a file; and  
providing the data in the group in the file contents clipboard format.

9. The method of claim 6 wherein the step of performing the data transfer comprises the step of using the clipboard to effect the data transfer.

10. The method of claim 6 wherein the step of performing the data transfer comprises the step of performing a drag-and-drop operation to effect the data transfer.

11. A computer system comprising:

(a) means for establishing a file group descriptor clipboard format for holding a file group descriptor;  
(b) a data source for providing the file group descriptor in the file group descriptor clipboard format, wherein said file group descriptor holds the file descriptors

with descriptive information about a group of data that are not files;

(c) a data sink for accepting transferred data in the file group descriptors clipboard format; and

(d) a data transfer mechanism for transferring the file group descriptor to the data sink.

12. The computer system of claim 11, further comprising:

(e) means for establishing a file contents clipboard format for holding contents of a file.

13. The computer system of claim 12 wherein the data source further comprises means for providing the group of data in the file contents clipboard format and the data transfer mechanism further comprises means for transferring the group of files in the file contents clipboard format from the data source to the data sink to create a group of files.

14. The computer system of claim 11 wherein the data transfer mechanism comprises a clipboard.

15. In a computer system having a clipboard for performing data transfers of data in clipboard formats, a method comprising the steps of:

establishing a file list clipboard format for storing a value for accessing a file list structure that describes a list of files involved in a data transfer operation;

providing data in the file list clipboard format; and

performing a data transfer on the data in the file list clipboard format.

16. The method of claim 15 wherein the step of providing data in the file list clipboard format comprises the

step of providing a value for accessing a file in the file list clipboard format.

17. The method of claim 15 wherein the step of providing data in the file list clipboard format comprises the step of providing an identifier to a list of files, said identifier being in the file list clipboard format.

18. The method of claim 15 wherein the step of performing the data transfer comprises the step of using the clipboard to perform a data transfer on the data in the file list clipboard format.

19. The method of claim 15 wherein the step of performing the data transfer comprises the step of performing a drag-and-drop operator to effect a data transfer or the data in the file list clipboard format.

20. A computer system comprising:

(a) means for establishing a file list clipboard format for holding a value for accessing a file list structure that describes a list of files involved in a data transfer operation;

(b) a data source for providing data in the file list clipboard format;

(c) a data sink for accepting data in the file list clipboard format; and

(d) a data transfer mechanism for transferring the data from the data source to the data sink.

21. In a computer system having an output device and a clipboard for performing data transfer of data in clipboard formats, a method comprising the steps of:

establishing an object positions clipboard format for storing relative positions of graphical objects to each other when displayed on the output device;

providing objects with associated graphical objects;

providing relative object positions for a current location of the graphical objects as displayed on the output device in the objects positions clipboard format; and

performing data transfer of the objects and the relative object positions to display the graphical objects at a new location on the output device with the same relative object positions.

22. The method of claim 22 wherein the data transfer is performed by the clipboard.

23. A computer system comprising:

(a) a data source for providing a number of objects;

(b) an output device for displaying graphical objects representing the objects provided by the data source;

(c) means for establishing an object positions clipboard format for storing relative positions of a group of graphical objects on the output device;

(d) a data transfer mechanism for transferring the objects from a data source to a data sink and for receiving, from the data object, relative object positions of the graphical objects in the object positions clipboard format so that the relative object positions of the graphical objects are preserved when the transferring is completed.

24. In a computer system having a clipboard for performing data transfers of data in clipboard formats, a method comprising the steps of:

establishing a filename map clipboard format for holding a list of at least one alternative name for items to be transferred via data transfer;

providing a list of an alternative name for each item in a list of items in the filename map clipboard format, the list of items including at least one item performing a data transfer on the list of items such that the item is

assigned its alternative name when the data transfer is complete.

25. The method of claim 24 wherein the data transfer is performed by the clipboard.

26. The method of claim 24 wherein the data transfer is effected by a drag-and-drop operation.

27. A computer system comprising:

- (a) means for establishing a filename map clipboard format for holding an alternative name for each item in a list of items;
- (b) a data source for providing a list of at least one item to be transferred;
- (c) means for providing a list of alternative names for the list of items provided by the data source such that an alternative name is provided for each item, said list of alternative names being in the filename map clipboard format;
- (d) a data sink for receiving the list of items to be transferred; and
- (e) a data transfer mechanism for transferring the list of items from the data source to the data sink such that each item in the list is assigned the alternative name provided in the list of alternative names that is in the filename map clipboard format.

28. In a computer system having a namespace of objects that includes file system objects and non-file system objects, and a clipboard for transferring data in clipboard formats; a method comprising the steps of:

establishing a selected clipboard format for holding data that enables access to be gained to the non-file system objects;

providing data that enables access to be gained to a given one of the non-file system objects in the selected clipboard format; and

performing a data transfer to transfer the given non-file system object using the data in the selected clipboard format.

29. The method of claim 25 wherein the given non-file system object is a network resource.

30. The method of claim 28 wherein the given non-file system object is a printer.

31. A computer system comprising:

(a) file system objects and non-file system objects;

(b) a namespace mechanism for providing a namespace for the file system objects and the non-file system objects;

(c) means for establishing a selected clipboard format for holding data that enable access to be gained to the non-file system objects;

(d) data transfer mechanism for transferring a given one of the non-file system objects using data that enables access to the given non-file system object, wherein the data is in the selected clipboard format.

32. In a computer system having a clipboard for transferring data in clipboard formats and a plurality of objects, where each object has an associated ID that identifies it, and ID's may be aggregated into ID lists, a method comprising the steps of:

establishing an ID list clipboard format for holding ID lists;

providing an ID list for a group of objects, said ID list holding ID's for each of the objects in the group and being in the ID list clipboard format; and

transferring the group of objects using a data transfer mechanism and the ID list in the ID list clipboard format.

33. A computer system comprising:

- (a) objects with ID's;
- (b) means for establishing an ID list clipboard format for holding an ID list having a list of ID's for some of the object;
- (c) a data source for providing an ID list in the ID list clipboard format for a group of objects; and
- (d) a data transfer mechanism for performing a data transfer on the ID list provided by the data source in the ID list clipboard format.

## 3. Detailed Description of Invention

Technical Field

The present invention relates generally to data processing systems and, more particularly, to data transfer within the data processing system.

Background of the Invention

The clipboard is a data transfer feature of the "MICROSOFT" WINDOWS, version 3.1, operating system sold by Microsoft Corporation of Redmond, Washington. The clipboard is used to transfer data between applications or within a single application. The clipboard includes a set of functions and messages that enable applications to transfer data via the clipboard. The clipboard may be viewed as a common area for storing data handles (i.e., unique identifiers for data objects) to which applications can exchange formatted data. The "MICROSOFT" WINDOWS, version 3.1, operating system enumerates a fixed number of clipboard formats. For example, the CF\_BITMAP clipboard format is used for transferring bitmaps, and the CF\_TEXT clipboard format is used for transferring arrays of text characters. The clipboard may simultaneously hold the same data in different clipboard formats.

Users of applications use the clipboard by calling clipboard commands, such as "cut", "copy" or "paste". The copy clipboard command copies a selected portion of data in a clipboard format to the clipboard by copying a handle to the data object that holds the selected portion of data in a clipboard format to the clipboard. The selected portion of data is not removed from the source from which it originated. The cut clipboard command is similar to the copy command in that it copies a selected portion of data to the clipboard, but the cut clipboard command differs from the copy clipboard command in that it removes the selected portion of data from the source of the data. The paste clipboard command copies data from the clipboard to a destination.

The Microsoft OLE 2.01 protocol, established by Microsoft Corporation, provides a mechanism for facilitating drag and drop operations. This mechanism uses clipboard formats. The data, however, is not passed through the system store that is used for cut and copy operations; instead, the transfer is directed from source to destination.

Although the clipboard is useful, the formatting constraints are too limiting for many applications. In particular, the number of clipboard formats provided by the system is too few, and the available formats are too limited.

#### Summary of the Invention

The present invention overcomes the limitations of the conventional system by providing expanded clipboard formats. These expanded clipboard formats may include a clipboard format for holding the contents of a file so that data, that is not a file, may be transferred and as part of the transfer, the data is encapsulated into a file. The expanded clipboard formats may also include a clipboard format for holding a file group descriptor. The file group descriptor holds a number of file descriptors and each file descriptor holds descriptive information about a file or about data that is to be incorporated into a file during a data transfer operation.

The expanded clipboard formats provided by an embodiment of the present invention may include a file list clipboard format for storing a value for accessing a file list structure. The file list structure describes a list of files. This file list clipboard format, like other clipboard formats, may be used during data transfer operations.

The expanded clipboard formats may also include an object positions clipboard format for storing relative positions of graphical objects when displayed on an output device. This clipboard format may be used for data transfers of objects to preserve the relative object positions of the graphical objects that are associated with the objects after the data transfer operation is completed.

The expanded clipboard formats may also include a filename map clipboard format for holding a list of alternative names for items that are being transferred during a data transfer operation. A list of alternative names held in the filename map clipboard format may be used to rename the files once the data transfer is completed.

Expanded clipboard formats provided by embodiments of the present invention may also include selected clipboard formats for holding data that enables access to non-file system objects. Certain computer systems may include file system objects and non-file system objects within a common namespace. This expanded clipboard format facilitates data transfers of the non-file system objects in the selected clipboard format. The non-file system objects may include network resources, printers, or other types of objects.

The expanded clipboard formats provided by embodiments of the present invention may include an ID list clipboard format for holding an ID list of objects. This clipboard format is useful in computer systems that have objects with associated ID's that may be aggregated into an ID list. The ID list clipboard format is useful when objects are transferred using a data transfer mechanism.

#### Detailed Description of the Invention

The preferred embodiment of the present invention provides an expanded number of clipboard formats. These expanded clipboard formats include clipboard formats that provide improved functionality over those provided by conventional systems.

Figure 1 is a block diagram of a computer system 10 that is suitable for practicing the preferred embodiment of the present invention. The computer system 10 includes central processing unit (CPU) 12 that has

access to several peripheral devices, including a video display 14, a mouse 16, and a keyboard 18. The CPU 12 also has access to a primary memory 20 and a secondary memory 22. The primary memory 20 holds an operating system 24 that includes clipboard 26. The clipboard 26 encompasses messages and functions for providing the clipboard capabilities to realize data transfer and the clipboard formats for practicing the preferred embodiment of the present invention. The primary memory 20 also holds code for at least one application program 28.

Those skilled in the art will appreciate that the computer system 10 of Figure 1 is intended to be merely illustrative. The present invention may also be practiced in other computer system configurations, including those that employ multiple processors and those that employ peripheral devices that differ from the devices shown in Figure 1.

The discussion below will focus on enumerating the expanded clipboard formats and their use as provided by the preferred embodiment of the present invention. It should be appreciated that these expanded clipboard formats supplement rather than supersede those provided by the "MICROSOFT" WINDOWS, version 3.1, operating system.

Figure 2 is a flowchart illustrating the steps that are performed to utilize the expanded clipboard formats of the preferred embodiment. Initially, an application program 28 must convert a selected portion of data into one or more of the expanded clipboard formats (step 30). The application program provides the mechanism for selecting the data that is to be converted into the expanded clipboard format. The converted data is stored in a data object (step 32). The data object may be in different types of storage mediums (such as provided by "MICROSOFT" OLE 2.01), including global memory, a stream or a storage. Data objects for some of the expanded clipboard formats may appear in only a subset of the possible storage mediums, as will be described in more

detail below. A handle to a data object that encapsulates the data structure is passed to the clipboard (step 34). The data is then considered to be on the clipboard in the expanded clipboard format and is pasted to a destination as desired by the user or application program (step 36).

One of the expanded clipboard formats provided by the preferred embodiment of the present invention is the CP\_HDROP clipboard format. For purposes of explaining the role of this clipboard format, it is helpful to first review how a drag-and-drop operation is performed in the "MICROSOFT" WINDOWS, version 3.1, operating system. In this conventional operating system, the drag-and-drop operation is performed in response to a user using a mouse. When the mouse button is released to perform a drop, the window in which the mouse cursor currently points receives a WM\_DROPOFILES window message. This window message contains a single parameter, hdrop. The hDrop parameter is a handle to a data structure that describes the file or files that have been dropped. The handle is used as a parameter in calls to the predefined DragQueryPoint(), DragQueryFile(), and DragFinish() functions that retrieve information from the data structure identified by hdrop.

The CP\_HDROP clipboard format is used to hold the handle to a DROPOFILES structure. The CP\_HDROP clipboard format requires that the data object (i.e., DROPOFILES) be in global memory. The DROPOFILES structure has the following format:

```
typedef struct _DROPOFILES {
    WORD pFiles;      // offset of file list
    POINT pt;         // drop point (client coords)
    WORD fNC;         // is it in non client area
    // and pt is in screen coords
    BOOL fInside;     // WIDE character switch
} DROPOFILES, *PDROPOFILES;
```

The fWide field indicates that the strings referenced by pfiles are in UNICODE rather than ANSI.

The DragQueryPoint() function, the DragQueryFile() function, and the DragFinish() function all may act upon the DROPTILES structure to retrieve information about dropped files. The use of the CF\_HDROP clipboard format enables the above-described functions to operate for clipboard operations as well as OLE data transfer operations. In conventional systems, these functions operate only in drag-and-drop situations.

The expanded clipboard formats of the preferred embodiment of the present invention also include the CF\_FILECONTENTS clipboard format and the CF\_FILEGROUPDESCRIPTION clipboard format. The CF\_FILECONTENTS clipboard format is used to hold data that is to be encapsulated into a file. For example, suppose that a user wishes to drag an embedding out of a mail message or other compound document and drop it on a location to create a file. In such an instance, the steps shown in the flowchart of Figure 3 are performed. First, the data to be transferred is selected (step 37). The data is converted into the CF\_FILECONTENTS clipboard format and stored (step 38). The data transfer operation is then completed (step 39) so that the target object of the drag-and-drop has access to the data that is stored in the CF\_FILECONTENTS clipboard format. The data in the CF\_FILECONTENTS clipboard format is then encapsulated into a file (step 40). Although this example was drag and drop, it should be appreciated that CF\_FILECONTENTS may be used with other types of data transfer operations.

In order to appreciate the role that clipboard formats serve in data transfer operations with the Microsoft OLE 2.01 protocol, it is helpful to review some fundamental concepts of OLE 2.01. An "object," in OLE 2.01, is a logical structure that includes data structures for holding data and may include functions that operate on the data held in the data structure.

Another concept that is fundamental to OLE 2.01 is the notion of an "interface." An interface is a name set of logically related functions. An interface lists signatures (such as parameters) for a set of functions. An interface does not provide code for implementing the functions; rather, the code for implementing the functions is provided by objects. Objects that provide the code for implementing the functions of an interface are said to "support" the interface. The code provided by an object that supports the interface must comply with the signature provided with the interface. In the C++ program language, an interface constitutes a set of virtual functions. The Microsoft OLE 2.01 protocol defines an IDataObject interface that group several functions which are useful for an object to transfer data. Objects that support the IDataObject interface are known as data objects.

During a drag-and-drop operation in which data is dragged from a source object to a target object in a system that uses OLE 2.01 to effect the data transfer, a uniform data transfer mechanism as defined in co-pending application entitled "Uniform Data Transfer," which was filed on February 22, 1994, Serial No. 08/199,853, and assigned to a common assignee with the present application, is utilized. The connection between a data object at the data source and the target object is established by passing a pointer for an instance of the IDataObject interface provided by the data object to the destination object. The destination object then executes the EnumFormatEtc() function of the IDataObject interface to enumerate the available formats for the data in the object that has been dropped on the destination object. These formats are clipboard formats. Thus, during drag-and-drop operations, the data is transferred in a clipboard format.

The CF\_FILEGROUDEDSCRIPTOR clipboard format is used to hold a file group descriptor structure. The file group descriptor structure holds one or more file

descriptors. Each file descriptor is an array of structures that hold information about data to be encapsulated into a file that is held in the CF\_FILECONTENTS clipboard format. A file descriptor has the following format:

```
typedef struct _FILEDESCRIPTOR { // fcd
    DWORD dwFlags;
    CLSID clsid;
    SIZEL sizeL;
    POINTL pointL;

    DWORD dwFileAttributes;
    FILETIME ftCreationTime;
    FILETIME ftLastAccessTime;
    FILETIME ftLastWriteTime;
    DWORD nFileSizeHigh;
    DWORD nFileSizeLow;
    CHAR cFileName[MAX_PATH];
} FILEDESCRIPTOR, *LPFILEDESCRIPTOR;
```

The dwFlags field is a file that indicates which fields hold legal data; the clsid field holds a class ID of the data object that encapsulates the CF\_FILECONTENTS data; the sizeL field holds a value specifying the size of the CF\_FILECONTENTS data; and the pointL field holds a pointer to the object that holds the CF\_FILECONTENTS data. The dwFileAttributes field holds a double word of attributes of the data. The ftCreationTime field holds a creation time, the ftLastAccessTime field holds a last access time, and the ftLastWriteTime holds a last write time. The nFileSizeHigh field holds the high 32 bits of a 64 bit value that describes the length of the object in bytes, and the nFileSizeLow field holds the low 32 bits of this value. The cFileName field holds characters that specify a filename.

The **CF\_FILEGROUPEDSCRIPTOR** clipboard format facilitates transferring multiple files in one batch. A file group descriptor in the batch for the batch to be transferred is stored in the **CF\_FILEGROUPEDSCRIPTOR** clipboard format. The file group descriptor must be in global memory and not other source media. This file group descriptor enables a program to walk through the data that is to be transferred into multiple files and create the resulting files as needed. Figure 4 is a flowchart illustrating the steps that are performed to use the **CF\_FILECONTENTS** clipboard format in conjunction with the **CF\_FILEGROUPEDSCRIPTOR** clipboard formats to transfer a group of data and create a group of files. The data to be encapsulated into the files must first be converted and stored in the **CF\_FILECONTENTS** clipboard format (step 41). A file group descriptor for the group of data is then stored in the **CF\_FILEGROUPEDSCRIPTOR** clipboard format (step 42). The data is transferred and encapsulated into a group of files (step 43).

It should be appreciated that the data objects for holding the data that is transferred in the **CF\_FILECONTENTS** clipboard format may be an object that is present in global memory or may be provided via OLE storage medium such as streams or storages.

The **CF\_FILENAMEMAP** clipboard format is another of the expanded clipboard formats provided by the preferred embodiment of the present invention. This clipboard format is used to provide storage for mappings to destination names. For example, suppose that a system wishes to change the name of a file when it is placed in a wastebasket facility. In such a case, the mappings to the destination names are stored in the **CF\_FILENAMEMAP** clipboard format. This clipboard format may be viewed as a companion to the **CF\_HDROP** clipboard format. Figure 5 is a flowchart illustrating the steps that are performed to utilize this clipboard format to change filenames. First, the destination name mappings are put into the

CF\_FILENAMEMAP clipboard format (step 44). The data transfer is then initiated via clipboard commands, drag-and-drop or other OLE data transfer mechanisms (step 46). As part of the completion of the data transfer operation, the new filenames specified within the data held in the CF\_FILENAMEMAP clipboard format are assigned to the files that are being transferred (step 48).

The expanded clipboard formats also include the CF\_OBJECTPOSITIONS clipboard format. This clipboard format is used to hold an array of coordinates that correspond to positions of other items in the clipboard. The first set of coordinates identifies a screen position of the group of objects and the remaining coordinates specify relative offsets of each item in pixels. This clipboard format is especially useful in transferring groups of objects that have associated icons. The clipboard format allows the preservation of the relative positioning of the group of items to remain the same after the data transfer is affected to a new destination.

Figure 6 is a flowchart illustrating the steps that are performed to utilize the CF\_OBJECTPOSITIONS clipboard format. Initially, the icons are encoded according to the CF\_OBJECTPOSITIONS clipboard format (step 50). The data transfer operation is then initiated via the clipboard, drag-and-drop or other OLE data transfer mechanisms (step 52). As part of the completion of the data transfer operation, the coordinates held in the clipboard format are used to realize the icons in appropriate relative positions at the destination (step 54).

The expanded clipboard formats further include a CF\_PRINTERFRIENDLYNAME clipboard format for holding a handle to a list of printer friendly names. The printer friendly names may then be used to gain access to data structures held for the corresponding printers. The CF\_NETRESOURCE clipboard format is similar but holds a handle to a list of network resources such as network

servers. The list of network resources may be used to gain access to data structures for the resources. This clipboard format is especially used for situations in which a name space may include not only file system structures but other types of objects.

The CF\_INDXLIST clipboard format holds a handle to a list of ID lists. An ID list is a list of identifiers that uniquely identify things within the name space. The ID list may be considered roughly analogous to pathnames for files or directories. This clipboard format may be viewed as a CF\_HDROP clipboard format.

While the present invention has been described with references to a preferred embodiment thereof, those skilled in the art will appreciate that various changes in forms of the detail may be made without departing from the intended scope of the present invention as defined by the appended claims.

## 4. Brief Description of Drawings

A preferred embodiment of the present invention will be described in more detail below with reference to the following figures.

Figure 1 is a block diagram of a computer system that is suitable for practicing the preferred embodiment of the present invention.

Figure 2 is a flowchart illustrating steps that are performed to exploit the expanded clipboard formats of the preferred embodiment of the present invention.

Figure 3 is a flowchart illustrating the steps that may be performed to use a CF\_FILECONTENTS clipboard format in a data transfer operation per the preferred embodiment of the present invention.

Figure 4 is a flowchart illustrating the combined use of the CF\_FILECONTENTS and CF\_FILEGROUNDESCRIPTOR clipboard formats in accordance with the preferred embodiment of the present invention.

Figure 5 is a flowchart illustrating the steps that are performed in an exemplary use of the CF\_FILENAMEMAP clipboard format in accordance with the preferred embodiment of the present invention.

Figure 6 is a flowchart illustrating the steps that are performed in exemplary use of the CF\_OBJECTPOSITIONS clipboard format in accordance with the preferred embodiment of the present invention.

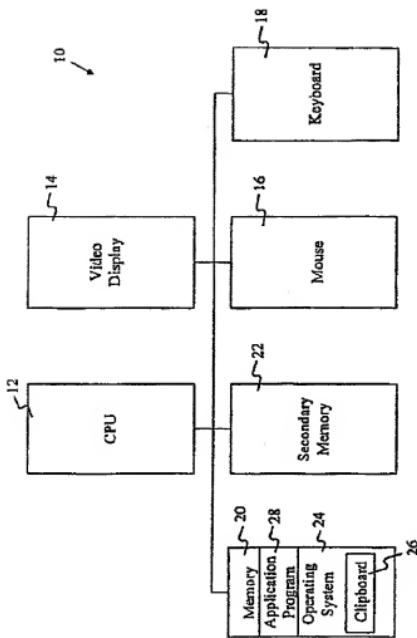
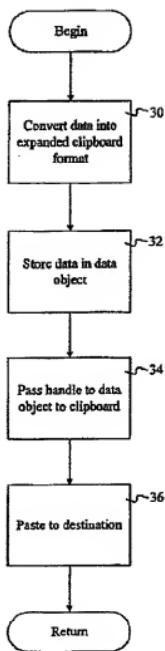
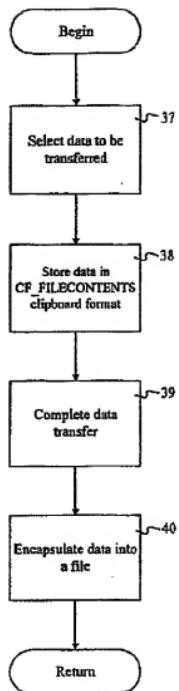


FIG. 1



*FIG. 2*



**FIG. 3**

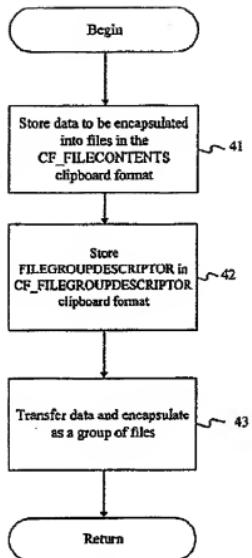


FIG. 4

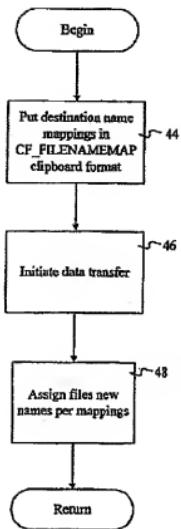


FIG. 5

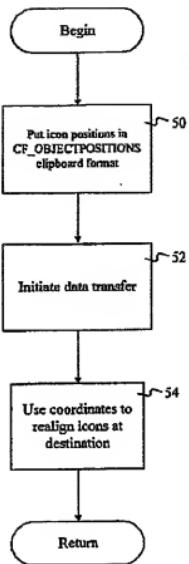


FIG. 6

**1. Abstract**

A computer system provides expanded clipboard formats that embellish the number of formats that may be used with a clipboard. These expanded clipboard formats enable users in applications to broaden their use of the clipboard and other data transfer mechanisms. The clipboard formats may be utilized by a conventional clipboard, by drag-and-drop mechanisms and by OLE data transfer mechanisms. Certain of the expanded clipboard formats are adapted for use in the data transfer of non-file system objects.

**2. Representative Drawing****Fig. 1**